AMENDMENT TO THE TITLE:

Please amend the Title to Invention to read as follows:

-- A RELATIONAL DATABASE MANAGEMENT SYSTEM (RDBMS) EMPLOYING A RELATIONAL DATASTORE AND A MULTI-DIMENSIONAL DATABASE (MDDB) FOR SERVING QUERY STATEMENTS FROM CLIENT MACHINES

AMENDMENT OF THE SPECIFICATION:

Please amend the Specification as follows:

On Page 1, please amend the last paragraph as follows:

Data warehousing, the creation of an enterprise-wide data store, is the first step towards managing these volumes of data. The Data Warehouse is becoming an integral part of many information delivery systems because it provides a single, central location where a reconciled version of data extracted from a wide variety of operational systems is stored. Over the last few years, improvements in price, performance, scalability, and robustness of open computing systems have made data warehousing a central component of Information Technology CIT strategies. Details on methods of data integration and constructing data warehouses can be found in the white paper entitled "Data Integration: The Warehouse Foundation" by Louis Rollleigh and Joe Thomas.~~, published at http://www.acxiom.com/whitepapers/wp-11.asp .~~

On Page 6, please amend the first full paragraph as follows:

Typically, in MDDB systems, the aggregated data is very sparse, tending to explode as the number of dimension grows and dramatically slowing down the retrieval process (as described in the report entitled "Database Explosion: The OLAP Report", ~~http://www.olapreport.com/DatabaseExplosion.htm~~ , incorporated herein by reference). Quick and on line retrieval of queried data is critical in delivering on-line response for OLAP queries. Therefore, the data structure of the MDDB, and methods of its storing, indexing and handling are dictated mainly by the need of fast retrieval of massive and sparse data.

On Page 8, amend the first paragraph as follows:

Modern operational and informational database systems, as described above, typically use a database management system (DBMS) (such as an RDBMS system, object database system, or object/relational database system) as a repository for storing data and querying the data. FIG. 14 illustrates a data warehouse-OLAP domain that utilizes the prior art approaches described above. The data warehouse is an enterprise-wide data store. It is becoming an integral

part of many information delivery systems because it provides a single, central location wherein a reconciled version of data extracted from a wide variety of operational systems is stored. Details on methods of data integration and constructing data warehouses can be found in the white paper entitled “Data Integration: The Warehouse Foundation” by Louis ~~Rollleigh~~ Rolleigh and Joe Thomas~~, published at http://www.acxiom.com/whitepapers/wp-11.asp~~.

On Page 11, amend the first paragraph as follows:

The first performance issue arises from computationally intensive table scans that are performed by a naive implementation of data joining. Indexing schemes may be used to bypass these scans when performing joining operations. Such schemes include B-tree indexing, inverted list indexing and aggregate indexing. A more detailed description of such indexing schemes can be found in "The Art of Indexing", Dynamic Information Systems Corporation, October 1999~~, available at http://www.disc.com/artindex.pdf~~. All of these indexing schemes replaces table scan operations (involved in locating the data elements that match a query) with one ~~ore~~ or more index lookup operation. Inverted list indexing associates an index with a group of data elements, and stores (at a location identified by the index) a group of pointers to the associated data elements. During query processing, in the event that the query matches the index, the pointers stored in the index are used to retrieve the corresponding data elements pointed therefrom. Aggregation indexing integrates an aggregation index with an inverted list index to provide pointers to raw data elements that require aggregation, thereby providing for dynamic summarization of the raw data elements that match the user-submitted query.

On Page 12, amend the third paragraph as follows:

In addition, systems that exploit hardware and software parallelism have been developed that lessens the performance issues set forth above. Parallelism can help reduce the execution time of a single query (speed-up), or handle additional work without degrading execution time (scale-up). ~~).~~ For example, Red Brick™ has developed STARjoin™ technology that provides high speed, parallelizable multi-table joins in a single pass, thus allowing more than two tables can be joined in a single operation. The core technology is an innovative approach to indexing

that accelerates multiple joins. Unfortunately, parallelism can only reduce, not eliminate, the performance degradation issues related to the star schema.

On Page 14, amend the second full paragraph as follows:

Accordingly, it is a further object of the present invention to provide an improved method of and system for managing data elements within a multidimensional database (MDDB) using a novel stand-alone (i.e. external) data aggregation server, achieving a significant increase in system performance (e.g. ~~deceased~~ decreased access/search time) using a stand-alone scalable data aggregation server.

On Page 16, amend the eighth paragraph as follows:

In another aspect, it is an object of the present invention to provide an improved method of and system for joining and aggregating data elements integrated within a database management system (DBMS) using a non-relational multi-dimensional data structure (MDDB), achieving a significant increase in system performance (e.g. ~~deceased~~ decreased access/search time), user flexibility and ease of use.

On Page 33, amend the first and second paragraphs as follows:

It should be noted that the DBMS typically includes additional components (not shown) that are not relevant to the present invention. The query interface and query handler service user-submitted queries (in the preferred embodiment, SQL query statements) forwarded, for example, from a client machine over a network as shown. The query handler and relational data store (tables and meta-data store) are operably coupled to the MDD Aggregation Module. Importantly, the query handler and integrated MDD Aggregation Module operate to provide for dramatically improved query response times for data aggregation operations and drill-downs. Moreover, it is an object of the present invention ~~is~~ to make user-querying of the non-relational

MDDB no different than querying a relational table of the DBMS, in a manner that minimizes the delays associated with queries that involve aggregation or drill down operations. This object is enabled by providing the novel DBMS system and integrated aggregation mechanism of the present invention.

FIG. 19B shows the primary components of an illustrative embodiment of the MDD Aggregation Module of FIG. 19A, namely: a base data loader for loading the directory and table(s) of relational data store of the DBMS; an aggregation engine for receiving dimension data and atomic data from the base loader, a multi-dimensional database (MDDB); a MDDB handler and an SQL handler that operate cooperatively with the query handler of the DBMS to provide users with query access to the MDD Aggregation Module, and a control module for managing the operation of the components of the MDD aggregation module. The base data loader may load the directory and table(s) of the relational data store over a standard interface (such as OLDB, OLE-DB, ODBC, SQL, API, JDBC, etc.). In this case, the DBMS and base data loader include components that provide communication of such data over these standard interfaces. Such interface components are well known in the art. For example, such interface components are readily available from Attunity Corporation~~, http://www.attunity.com~~.

On Page 34, amend the last paragraph as follows:

In step 607, a reference is defined that provides users with the ability to query the data generated by the MDD Aggregation Module and/or stored in the MDDB of the MDD Aggregation Module. This reference is preferably defined using the Create View SQL statement, which allows the user to: i) define a table name (TN) associated with the MDDB stored in the MDD Aggregation Module, and ii) define a link used to route SQL statements on the table TN to the MDD Aggregation Module. In this embodiment, the view mechanism of the DBMS enables reference and linking to the data stored in the MDDB of the MDD Aggregation Engine as illustrated in FIG. ~~6(E)~~ 19E. A more detailed description of the view mechanism and the Create View SQL statement may be found in C.J. Date, "An Introduction to Database Systems," Addison-Wesley, Seventh Edition, 2000, pp. 289-326, herein incorporated by reference in its entirety. Thus, the view mechanism enables the query handler of the DBMS system to forward any SQL query on table TN to the MDD aggregation module via the

associated link. In an alternative embodiment, a direct mechanism (e.g., NA trigger mechanism) may be used to enable the DBMS system to reference and link to the data generated by the MDD Aggregation Module and/or stored in the MDDB of the MDD Aggregation Engine as illustrated in FIG. ~~6F~~ 19F. A more detailed description of trigger mechanisms and methods may be found in C.J. Date, "An Introduction to Database Systems," Addison-Wesley, Seventh Edition, 2000, pp. 250, 266, herein incorporated by reference in its entirety.

On Page 35, amend the second full paragraph as follows:

In step 611, the query handler receives the SQL statement(s); and optionally transforms such SQL statement(s) to optimize the SQL statement(s) for more efficient query handling. Such transformations are well known in the art. For example, see Kimball, "Aggregation Navigation With (Almost) No MetaData", DBMS Data Warehouse Supplement, August 1996~~, available at http://www.dbmsmag.com/9608d54.html~~.